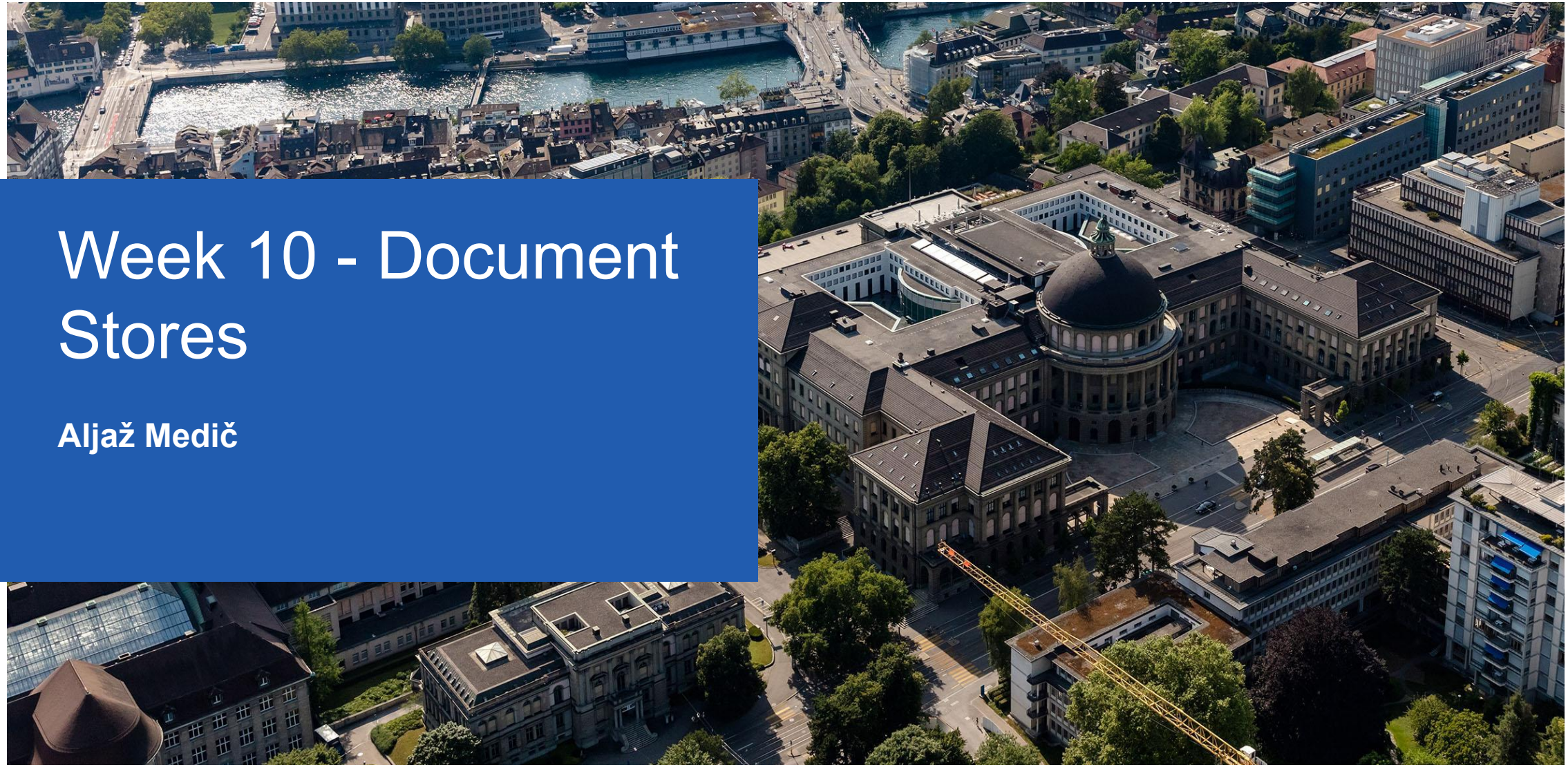
ETH zürich



Week 10 - Document Stores

Aljaž Medič

Plan for today

1. Finishing last week's topic: SQL, Spark DF & SparkSQL
2. Correction of Quiz 9
3. Quiz
4. Mongo queries

Programming SQL

Using the discogs database available via the Jupyter notebook, write a SQL query and answer the following question.

What is the name of the artist who has the highest number of releases in the country "Switzerland"?

Notes:

1. "Various Artists" is not considered an artist.

2. One release_id is considered as one release.

3. The solution is unique. The result must be given by copy-pasting the name, without any extra whitespace or character.

Antwort:

All 3 Spark APIs (HS22, Q46)

With the orders.jsonl dataset available via the Jupyter notebook, use either the DataFrame API or Spark SQL and answer the following question.

How many unique customers ordered the product "stuffed animal"?

Hints:

1. We assume customers are uniquely identified with their first and last names.

2. The product "stuffed animal" means that the field product has the value "stuffed animal".

3. You can import the function "explode" with: from pyspark.sql.functions import explode.

4. The result must be given as an integer. No extra spaces, decimal periods, commas or superfluous zeros are allowed. Example: 0, 2, 42, 930, 3456, but NOT 03, 3.0, 4,2 or 1 234.

All 3 Spark APIs (HS22, Q48)

With the orders.jsonl dataset available via the Jupyter notebook, use either the DataFrame API or Spark SQL and answer the following question.

What is the date at which the order with the largest quantity was placed?

Hints:

1. If you want to use the DataFrames API, then use `from pyspark.sql.functions import desc` to import sorting in descending order.

2. The date must be entered in the format YYYY-MM-DD, for example February 12, 2022 must be entered as 2022-02-12 but NOT 2022-2-12 or 2022-12-02.

Quiz 9 Correction:

Assignment 4

Focus on the first playing day of August 2013. Which language had the worst accuracy?

(In case of a tie, return the language with more games played.)

Assignment 5

Return number of games where the guessed language is correct and it appeared last in the choices list.

Hint: Check the docs of function element_at

QUIZ

What is the biggest benefit of document stores?

The ability to natively handle schema-less and nested data.

We can optionally validate data after the collection has already been populated.

True. (We have schema-on-read.)

MongoDB stores JSON internally as an optimized binary format.

True. (BSON)

QUIZ

MongoDB offers a high-level query language.

False. (We have an API language, no declarative queries.)

MongoDB assigns a unique ObjectID to all the documents, which is by default 12 bytes long.

True.

MongoDB is optimized for joining data across collections.

False. (Joins are supported, but discouraged.)

QUIZ

Quoting keys while using MongoDB through node.js is optional.

True.

MongoDB can sort across all types.

True. (null, numbers, strings, objects, arrays, binary data, object IDs, Booleans, dates, TS, regex)

In MongoDB, `.find` function performs a projection.

False. (It performs a selection.)

QUIZ

In SQL we can project away with `~column`; In MongoDB, we project-away with `{column: 0}`.

False. (SQL doesn't have project-away. MongoDB, however, does.)

We are not allowed to mix 0s and 1s in the projection, except for the `_id` field.

True.

The ordering of functions we use to build the query on top of `.find()` is important.

False. (It matters only in aggregation pipelines.)

QUIZ

MongoDB documents are generally larger than the HDFS blocks (64-128MB).

False. (Usually capped between 10-16MB, due to performance reasons.)

We can create indexes on multiple fields using B+ trees.

True.

A compound index can be used efficiently for queries involving any subset of indexed fields.

False. (Only for prefixes of the compound index.)

MongoDB Queries

For each of the following queries, state whether it is valid or not.

```
db.animals.find({$not:{"age":{$gt: 3}}})
```

Invalid. ($not ordering: `{"age":{$not:{$gt:3}}}`)

```
db.animals.find({"age":{$gt: 3, $lte: 8}})
```

Valid.

```
db.animals.find({"species": "dog", "hasOwner": True})
```

Invalid. (Watch out! python syntax.)

```
db.animals.update({"$set":{"age": 5}})
```

Invalid. (Missing filter argument: `.update(filter, update, …)`)

Old exam questions

For each of the following queries, state whether the index helps or not.

```
{
        "_id": 1245,
        "name": "Buddy",
        "species": "dog",
        "registration_date": "2019-04-03",
        "owners": [{
                "firstname": "John",
                "lastname": "Doe",
                "age": 25,
                "address": "Fake Street 23, Fake City"
        }, {
                "firstname": "Louis",
                "lastname": "Lane",
                "age": 30,
                "current_owner": true
        }]
}
```

IDX: `createIndex({"species": 1,"name":1})`

Query: `find({"name":"Bobby"})`

Not useful.

IDX: `createIndex({"name": 1, "species":-1})`

Query: `find({}).sort({"name":-1, "species":1})`

Useful.

Old exam questions

For each of the following queries, state whether the index helps or not.

```
{
        "_id": 1245,
        "name": "Buddy",
        "species": "dog",
        "registration_date": "2019-04-03",
        "owners": [{
                "firstname": "John",
                "lastname": "Doe",
                "age": 25,
                "address": "Fake Street 23, Fake City"
        }, {
                "firstname": "Louis",
                "lastname": "Lane",
                "age": 30,
                "current_owner": true
        }]
}
```

IDX: `createIndex({"owners.firstname": 1})`

Query: `find({"owners":{$elemMatch: {"firstname":"Jane", "lastname":"Doe"}}})`

Useful.

IDX: `createIndex({"owners": 1})`

Query: `find({"owners":{"firstname":"Jane", "lastname":"Doe"}})`

Useful.

Old exam questions

For each of the following queries, state whether the index helps or not.

```
{
        "_id": 1245,
        "name": "Buddy",
        "species": "dog",
        "registration_date": "2019-04-03",
        "owners": [{
                "firstname": "John",
                "lastname": "Doe",
                "age": 25,
                "address": "Fake Street 23, Fake City"
        }, {
                "firstname": "Louis",
                "lastname": "Lane",
                "age": 30,
                "current_owner": true
        }]
}
```

IDX: `createIndex({"name": 1, "species":-1})`

Query: `find({"name":"Alan"}).sort({"species":-1})`

Useful.

IDX: `createIndex({"name": 1, "species":-1})`

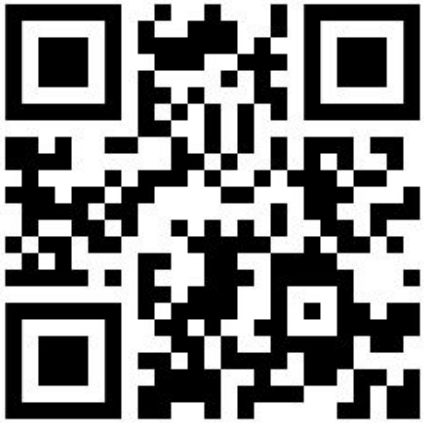
Query: `find({}).sort({"species":1})`

Not useful.

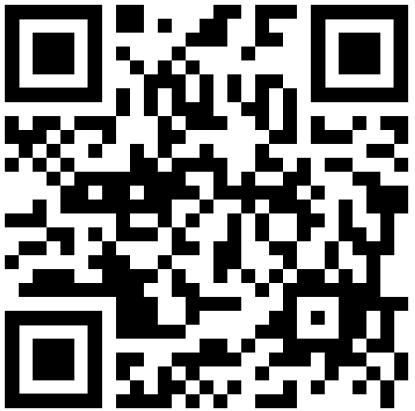
ETH zürich

See you next week!

Aljaž Medič
amedic@ethz.ch



Slides



Suggestions